Local Search for Optimal Global Map Generation Using Mid-Decadal Landsat Images

Lina Khatib[1,2] and **John Gasch**[3] and **Robert Morris**[2] and **Steven Covington**[4]
[1] PSGS
[2] Computational Sciences Division, NASA Ames Research Center
[3] Landsat Mission Operations, Goddard Space Flight Center
[4] USGS Landsat Flight Systems Manager
Landsat Project International Coordinator
Goddard Space Flight Center

Abstract

NASA and the US Geological Survey (USGS) are seeking to generate a map of the entire globe using Landsat 5 Thematic Mapper (TM) and Landsat 7 Enhanced Thematic Mapper Plus (ETM+) sensor data from the "mid-decadal" period of 2004 through 2006. The global map is comprised of thousands of scene locations and, for each location, tens of different images of varying quality to chose from. Furthermore, it is desirable for images of adjacent scenes be close together in time of acquisition, to avoid obvious discontinuities due to seasonal changes. These characteristics make it desirable to formulate an automated solution to the problem of generating the complete map. This paper formulates a Global Map Generator problem as a *Constraint Optimization Problem* (GMG-COP) and describes an approach to solving it using local search. Preliminary results of running the algorithm on image data sets are summarized. The results suggest a significant improvement in map quality using constraint-based solutions.

Introduction and Motivation

NASA and USGS are partnering to produce high quality image maps of the Earth's landmass using Landsat 5 Thematic Mapper (TM) and Landsat 7 Enhanced Thematic Mapper Plus (ETM+) sensor data from the mid-decadal period of 2004 through 2006. The map will be composed of roughly 9500 WRS[1] Landsat scene locations for which there are often tens of images available to select from. Eventually, over 300,000 images must be evaluated and down-selected to create the final survey data set. The resulting data map will be distributed to the public at no charge through a USGS website. In addition to providing benefits to researchers in the Earth sciences, it will likely become the next generation backdrop for Google-Earth (which currently uses the GeoCover-2000 data set).

[1]Landsat 5 and 7 follow the Worldwide Reference System 2 (WRS-2) coordinate system for indexing locations on the Earth where data is acquired. WRS-2 indexes a location via a set of paths and rows, with a 16-day repeat cycle. Landsat 5 follows the WRS-2 system with a temporal offset of 8 days relative to Landsat 7. The WRS-2 indexes orbits (paths) and scene centers (rows) into a global grid system (daytime and night time) of 233 paths by 248 rows. We refer to each path, row element as a *scene location.*

A collection of diverse criteria defines a high quality image map. First, a good map will typically consist of the best (most cloud-free) image data available per scene. Second, each image is assigned a Normalized Difference Vegetation Index (NDVI) as a metric of the health and density of vegetation on a patch of land. For Earth Science applications, images with high NDVI are typically preferred. Third, to be usable for regional scientific studies, it is preferable to choose image data that are seasonally consistent with neighboring scenes. Fourth, to accommodate land-cover/land-use change analysis, consideration must be given to the seasonality of previous survey data sets. Finally, because of a malfunction in the image scanner on Landsat 7 since 2003, ETM+ produces imagery that has coverage discontinuities such that an individual image covers only 78% of the land area. To compensate, two images of the same scene taken on different days are combined to produce a composite image that partially or fully closes the gaps. Pairs of images of a common scene must therefore be chosen to maximize coverage (minimize gap), which means the two scenes should be mutually out of phase. Each image is assigned a "gap–phase statistic", or GPS, which is an absolute measure of the geometric registration of the image scan line with respect to the scene center point. Such GPS values are used to compute the area coverage of composite images.

The size of the solution space, as well as the number of criteria for quality, make it desirable to develop automated solutions to the problem of generating high quality global image maps. The map generation problem can be naturally viewed as a *Constraint Optimization Problem*. This paper presents a formulation of the map generation problem, and describes an approach to solving the problem using local search. Section 2 describes the criteria for evaluating the quality of solutions. Section 3 summarizes related work and complexity issues, and Section 4 describes a local search approach to solving the problem. Section 5 summarizes preliminary experimental results.

GMG-COP

Global Map Generation (GMG) can be viewed as a *Constraint Optimization Problem* (GMG-COP) (Larrosa & Dechter 2003), with a set of variables $V = \{v_{i,j}\}$ indexed by WRS-2 path and row number i, j. Each variable $v_{i,j}$ represents a scene location, and is associated with a *do-*

main $D_{i,j} = \{d_{i,j,1}, \ldots, d_{i,j,m}\}$, where each $d_{i,j,k}$ represents a TM or ETM+ image taken of the corresponding scene. There are binary links with the 4 neighboring scenes, designated $north(v_{i,j-1})$, $east(v_{i-1,j})$, $south(v_{i,j+1})$, and $west(v_{i+1,j})$. A solution is a complete assignment of images (a single TM or a pair of ETM+) for each variable.

The GMG problem is an example of a multiobjective optimization problem, in which a set of potentially competing preference criteria are used to evaluate and compare solutions. The preference criteria are the following:

1. Single image criteria:
   - Minimize cloud cover;
   - Maximize NDVI value;
   - Seasonality with previous data sets;
   - Relative preference for acquiring L7 versus L5 images.
2. ETM+ composite criteria:
   - Minimize gaps in data that remain from compositing image pairs;
   - Minimize the difference in the days the composited images were acquired.
3. Criteria relating pairs of adjacent images:
   - Minimize the difference in the days the images were acquired;
   - Minimize the difference between the days in the year (ignoring the year) the images were acquired.

Each image is represented as a vector of "meta-data" comprised of the following text fields: the WRS-2 scene path and row numbers, the sensor that acquired the image (TM or ETM+), the acquisition date, the cloud cover assessment, the NDVI metric, and a GPS value used for evaluating the scene area coverage that results from compositing two images. In addition, a preference date is given. This date is to be used in our future work to gear our solver toward solutions with minimum perturbations from these preferred dates.

Each meta-data element is associated with a function that is used to evaluate solution quality. We normalize by considering *merit* values in the range $[0, 1]$ where 0 is worst and 1 is best. This way the objective function is a maximization and always positive. The quality of an individual image can be depicted in terms of two functions, related to the NDVI merit value, and to cloud cover: $ndvi : D \rightarrow [0, 1]$, and $acca : D \rightarrow [0, 1]$. Second, there are two functions associated with measuring the time difference between the acquisition of pairs of images: *Absolute Day Difference*, $date\Delta : D \times D \rightarrow [0, 1]$ is the number of days between image acquisition, and *Day of Year Difference*, $doy\Delta : D \times D \rightarrow [0, 1]$ is the gap in days (ignoring the year in which it was acquired). The latter function is used to reward solutions that assign images that are seasonally similar, regardless of year, whereas the former rewards solutions with pairs of images taken in the same year. Third, the function *Area Coverage*, $cover : D \times D \rightarrow [0, 1]$ assigns a value that indicates goodness of fit between pairs of images used in a composite. Finally, to express relative preferences for TM or ETM+ images, the function $IsL5 : D \rightarrow \{0, 1\}$, $IsL7 : D \rightarrow \{0, 1\}$ assign 1 to images acquired by TM (respectively, ETM+), and 0 otherwise.

A *solution* s to the GMG COP is a set of assignments $s = \{v_{i,j} \leftarrow \langle d_{i,j,k}, d_{i,j,l} \rangle\}$, where by convention $d_{i,j,k}$ is the *base image* and $d_{i,j,l}$ is the *fill image*. The pair represents the composition of L7 images; thus if the base is a TM image, hence requiring no fill, we set $d_{i,j,l} = d_{i,j,k}$ by convention. For an arbitrary solution s, we write $b_s(v_{i,j}), f_s(v_{i,j})$ for the base and fill values for the scene $v_{i,j}$ assigned by s.

The WRS organization of scene locations into path and row induces a grid or lattice structure to the GMG-COP. Because of the symmetry of adjacency, it suffices to represent this notion in terms of the functions $north : V \rightarrow V$ and $east : V \rightarrow V$, which return the variable corresponding to the scene that is north (east) of the designated variable.

The set of solutions can be ordered in terms of the objective of maximizing individual scene quality while maximizing phase difference between bases and fills and minimizing the temporal differences between (the bases of) adjacent images. Given an arbitrary solution s, its score is the value of the following weighted summation:

$$
\begin{aligned}
f(s) \;=\; & \Sigma_{i,j} \\
& w_1 * ndvi(b_s(v_{i,j})) \\
+ \; & w_2 * acca(b_s(v_{i,j}))) \\
+ \; & w_3 * ndvi(f_s(v_{i,j})) \\
+ \; & w_4 * acca(f_s(v_{i,j})) \\
+ \; & w_5 * date\Delta(b_s(v_{i,j}), f_s(v_{i,j})) \\
+ \; & w_6 * cover(b_s(v_{i,j}), f_s(v_{i,j})) \\
+ \; & w_7 * date\Delta(b_s(v_{i,j}), b_s(north(v_{i,j}))) \\
+ \; & w_8 * date\Delta(b_s(v_{i,j}), b_s(east(v_{i,j}))) \\
+ \; & w_9 * doy\Delta(b_s(v_{i,j}), b_s(north(v_{i,j}))) \\
+ \; & w_{10} * doy\Delta(b_s(v_{i,j}), b_s(east(v_{i,j}))) \\
+ \; & w_{11} * IsL5(b_s(v_{i,j}) \\
+ \; & w_{12} * IsL7(b_s(v_{i,j})
\end{aligned}
$$

w_1 and w_2 govern the importance of the quality of individual base images. $w_3 - w_6$ discount the value of an image based on the quality of the fill, and on the goodness of fit between base and fill. For L5 images, where the base and fill are the same, the discount is the same as $(w_2 + w_4)acca(b_s(v_{i,j}))$, etc. Notice that since we assume that the temporal and spatial match between an image and itself is perfect, L5 images are not discounted on these criteria. $w_7 - w_{10}$ deal with compatibility of bases with adjacent images (we ignore the compatibilities of fill), and w_{11} and w_{12} allow for an absolute preference for L5 or L7 images to be expressed. An optimal solution s^* to this GMG problem is one that receives the maximum score based on this function.

Complexity

Computationally, the problem solved by GMG is similar in structure to the assignment of frequencies to radio transmitters (Cabon *et al.* 1999), and other generalizations of the map coloring problem. To evaluate the complexity of the GMG-COP it is relevant to consider recent theoretical

results pertaining to the transformation of inputs to optimization problems into tree-like structures and solving them by applying dynamic programming techniques. In cases in which it can be shown that a "width" parameter related to the acyclicity of the input graph is bounded for a class of problem, the solution can be found in polynomial time. This approach has been used to solve graph problems in general (Hicks, Koster, & Kolotoglu 2005) and in particular CSPs (Samer & Szeider 2006) and COPs (Larrosa & Dechter 2003).

Bucket elimination (Dechter 2003) (BE) is an example of this approach; it is a complete dynamic programming algorithm that has been applied to solving COPs. It is based on the process of incrementally replacing variables with constraints that summarize the effect of the variable on finding the optimal solution. Given an ordering of the variables, this processing occurs in reverse order. The worst case time and space complexity is tightly bounded by a parameter of the problem called the *induced width*, which arises out of an ordering of the variables. Specifically, complexity of bucket elimination is $O(n * d^{w+1})$, where n is the number of variables, d is the size of the largest domain, and w is the induced width. In practice, the primary drawback in performance is space; only problems with small induced width can be solved.

Given a set of variables and associated constraints, finding the ordering of the variables with a *minimum induced width* is an NP-hard problem. Although to our knowledge no proof exists, it appears that the induced width of a constraint graph arranged as a square grid of size $n \times n$ is n. This linear growth rate imposes a practical limitation on the size of problems Bucket Elimination to roughly $n = 30$, too small for the GMG problem (Larrosa 2007). Hybrid approaches that combine BE with Branch and Bound have demonstrated an improvement in performance over pure BE for problems with a grid structure (Larrosa, Morancho, & Niso 2005). Although these results may justify the application of these methods to the GMG problem, in this paper we limit our attention to local search.

Local Search Solution

Reasons for adopting a local search solution to constraint optimization problems are well documented: they include

1. *Anytime performance:* On average, local search behaves well in practice, yielding low-order polynomial running times (Aarts & Lenstra 1997). Because the criteria space is high-dimensional, it is difficult *a priori* to quantitatively characterize globally preferred solutions. Consequently, our customers were interested in a system that could examine large parts of the search space quickly to determine weight settings that produced adequate results.
2. *Flexibility and ease of implementation:* Our customers required us to build, and demonstrate the advantages of, automated solutions in a short period of time (2 months). Local search can be easily implemented.
3. *Ability to solve large problems:* As optimization problems go, the GMG-COP can be considered large. Local search has been shown to be effective on large problems.

Local search defines a class of approximation algorithms that can find near-optimal solutions within reasonable running times. (The Simplex Algorithm for solving Linear Programming problems is an example of local search.) Given the pair (S, f), where S is the set of solutions and f is the objective function, let $S*$ be the set of best solutions (i.e., the ones with the highest score according to f), and $f*$ be the best score. Members of $S*$ are called *global optima*. Local search iteratively searches through the set S to find a *local optimal* solution, a solution for which no better can be found. Local optima need not be in general members of $S*$.

To conduct the search, local search relies on the notion of a *neighborhood function*, $N : S \rightarrow S$, a function that takes one solution (called the *current solution*) and returns a new solution that differs from the current in some small way. To be effective, a neighborhood function should be simple; it should not require a lot of time to compute. For GMG-COP, the neighborhood function randomly selects a cell and replaces the selected image with a new one. A neighborhood function is *exact* if every local optima it finds as the result of search is a global optima. The neighborhood function used to solve GMG-COP is not exact.

Designing a local search algorithm is based on deciding three components: how an initial solution, or *seed* is generated, how to select a neighboring solution, and when to terminate search. For the GMG solver described in this paper, we took a simple approach to deciding these issues, reasoning that complexity should be introduced only as needed, i.e., only as warranted by inferior performance of simpler approaches.

First, a good design for a seed generator is one that intuitively *starts in a good location* in the search space. A good location is one that is relatively close to optimal solutions, where close is measured by the length of the path from it to an optimal solution using the neighborhood function. For the GMG-COP, we assumed that a good place to start a search should be a solution that picks the highest individual quality image for each cell, ignoring preferences related to adjacency.

Choosing a neighboring solution requires, first, choosing which cell to change. The simplest approach is to pick the cell at random. Since local search is "memoryless", in the sense that it does not keep track of where it's been previously, it may not be able in general to avoid examining the same solution multiple times. To avoid this, sometimes algorithms have "taboo" lists, lists of variables recently chosen to change. Variables are put on the list after chosen and eventually taken off after some number of iterations. Variables on the list can't be selected on a given iteration. In our implementation we applied an extreme case of "taboo" list: once a scene is selected for examination, it is immediately placed on the taboo list (conceptually) to allow for all other scenes to be examined in the current iteration (pass over all scenes in random order).

Given a selected cell, there are also a number of ways to select among the set of neighboring solutions based on changes made to that cell. Some are deterministic; i.e., given the same decision to make, the algorithm will make the same choice each time. Others are non-deterministic. Algo-

rithms such as simulated annealing and genetic algorithms are non-deterministic. Initially, we opted for a deterministic approach, of which there are two kinds: first improvement or best improvement. First improvement examines neighbors, in a local search sense, until one is found that is better than the current solution; that one becomes the new current solution. Best improvement examines all the neighbors, and picks the one that improves upon the current solution the most. Either of these generates a *greedy* approach, one that always chooses an improving solution. A variation of best improvement is where a neighbor with the best score is chosen, even if the score is worse than the score of the current solution. Notice that this variant is not strictly greedy. It is sometimes preferred because it allows for the possibility that a globally optimal solution may not be on the "greedy path" from an initial seed solution.

Finally, choosing a termination condition requires deciding how many solutions will be generated before the algorithm halts. The simplest approach will be to define a termination condition that says *halt when you reach the first locally optimal solution or after a fixed number of solutions, MAX, have been generated*, whichever comes first. A slightly more sophisticated version of this *simple local search* is called *multi-start*: here, for some fixed number of runs, we start with different initial (seed) solutions. Such initial solutions can be fully randomly generated (our implementation), semi-randomly generated, or deterministically generated. An example of deterministically generated initial solutions employed here is to assign to each scene the best self-quality image/pair. Alternatively, the local optimum of one run of simple local search can be used as the initial solution for the next run. A simple local search algorithm, and the first and best improvement routines embedded within it, are presented below. ($N(S, c)$ is the set of all solutions S' where only the assignment to the scene cell c is changed.)

Iterative first improvement (S)
$bestI = S$
Loop over all scene cells, c, in random order
  For each $S' \in N(S, c)$
    if $f(S') > f(bestI)$ $bestI = S'$; continue;
end loop
return $bestI$

Iterative best improvement (S)
$bestI = S$
Loop over all scene cells, c, in random order
  For each $S' \in N(S, c)$
    if $f(S') > f(bestI)$ $bestI = S'$;
end loop
return $bestI$

Simple Local search (MaxPasses)
$bestS = seedS$
For up to MaxPasses iterations do
  S' = Iterative first (or best) improvement ($bestS$)
  if $S' = bestS$ return $bestS$; (local minimum reached)
  else $bestS = S'$;
return $bestS$

Testing and Results

Testing the GMG-COP occurred in two stages. First, we were interested in the extent of the improvement offered by an automated solution over current practice, which consists of manually generating solutions. Towards this end, tests were conducted by the customers at USGS and the Landsat mission using the GeoCover-2000 (GC2K) data set. The results showed that GMG, implemented as a simple algorithm then which we later improved significantly, produced a solution that was 23% better quality than the manually generated solution, based on the objective function scores. The customers viewed this result as significant enough to warrant further experiments.

The purpose of the follow-on experiments was to compare different variations in multi-start local search to determine the best performing algorithm. Four different multi-start local search variants were tested, based on two variations of two criteria: the initial solution and the choice of neighbor. These are standard ways of varying local search, representing tradeoffs between exploration and exploitation of domain knowledge. The initial solutions tried were a randomly generated solution (R) and the solution consisting of the set of images that scored highest individually (i.e. with respect to cloud cover, NDVI, and base–fill quality) (S). The choice of neighbor was either done on a "first improvement" (F) basis, i.e., the first alternative that improved the overall score, or "best improvement" basis, i.e., of all the images, selecting the one that most improved the score (B). The result was 4 variations, called RB, RF, SB, and SF.

These experiments were conducted on a subset of the mid-decadal Landsat data set consisting of 197 scenes over North America. 10000 separate starts of each algorithm were conducted, with different random seeds.

The results are shown in Figure 1. The results indicate that the best strategy for finding high quality solutions is through exploration: with a random initial solution, and an exhaustive search for the best neighbor (RB), progress was quickly made towards solutions with higher quality than those found by the other approaches. Somewhat surprisingly, starting with a good solution (SB, SF) hindered the performance through failure to escape from local optima. Further, both performed about equally, indicating that the extra examination of neighbors conducted by the best-neighbor approach did not help. Less surprisingly, random initial solution with first best (RF) performed the worst, not exploring enough of the space of neighboring solutions.

Because this experiment was conducted on a single data set, the results are clearly preliminary. Future tests on larger and/or different subsets of the mid-decadal data set are being planned, as well as tests on other data sets.

Summary and Future Work

The global map generation problem provides an ideal domain for testing and evaluating constraint-based optimization solvers. Furthermore, the GMG solver is of significant potential benefit to the Earth Science research community, allowing scientists access to improved automated tools to study the Earth's changing eco-system.

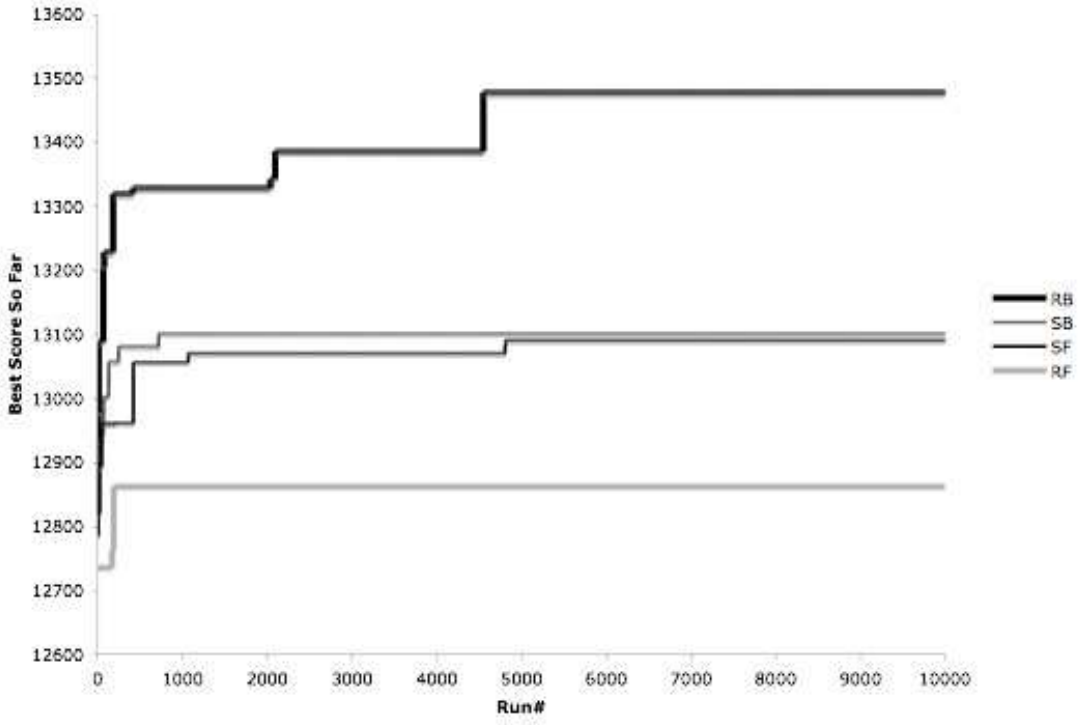

Figure 1: Comparison of performance of four versions of multi-start local search GMG: random initial solution with best improvement among neighbor scenes (RB), random initial solution with first improvement (RF), initial solution with highest individually scoring scenes with best improving (SB) and with first improving neighbor (SF). Best solution found so far at each of 10000 runs.

Based on a successful application of the GMG on the mid-decadal global Landsat data set, the GMG will then be used for additional data set projects as well. One such mapping application has to do with the USGS goal to create a state mosaic for all 50 states in the US. The use of GMG has the potential of automating a large part of that effort, viz., scene selection. Due to the scanner failure on Landsat 7, GMG will greatly reduce the labor necessary to exploit the Landsat data archive; and, as the Landsat 7 mission is expected to go to 2012, the benefit of the GMG cannot be overstated.

Despite satisfaction with the local search GMG, the customers at USGS and Landsat have expressed an interest in a version of GMG that uses a complete technique, i.e., that potentially examines the entire space of solutions. As noted earlier, a hybrid approach combining BE with Branch and Bound seems to be the most promising complete approach currently being explored. Future reports will focus on the results of applying complete approaches to solving GMG.

Acknowledgements

Thanks to Jeremy Frank for his valuable contributions to this work. Thanks also to anonymous reviewers for pointing out errors in the submitted draft.

References

Aarts, E., and Lenstra, J. K. 1997. *Local Search in Combinatorial Optimization*. John Wiley and Sons.

Cabon, B.; de Givry, S.; Lobjois, L.; Schiex, T.; and Warners, J. P. 1999. Radio link frequency assignment. *Constraints* 4(1):78–87.

Dechter, R. 2003. *Constraint Processing*. Morgan Kaufmann.

Hicks, I. V.; Koster, A.; and Kolotoglu, E. 2005. Branch and tree decomposition techniques for discrete optimization. *Tutorials in Operation Research, INFORMS* 1–29.

Larrosa, J., and Dechter, R. 2003. Boosting search with variable elimination in constraint optimization and constraint satisfaction problems. *Constraints* 8(3):303–326.

Larrosa, J.; Morancho, E.; and Niso, D. 2005. On the practical use of variable elimination in constraint optimization problems: 'still-life' as a case study. *JAIR* 23:421–440.

Larrosa, J. 2007. Javier Larrosa, personal correspondence.

Samer, M., and Szeider, S. 2006. Constraint satisfaction with bounded treewidth revisited. In Benhamou, F., ed., *Principles and Practice of Constraint Programming - CP2006*, 499–513.